DOCUMENT RESUME

ED 052 252 TM 000 658

AUTHOR	Mendro, Robert
TITLE	A Procedure for the Simulation of Test Item Score Distributions.
INSTITUTION	Illinois Univ., Urbana. Dept. of Educational Psychology.
PUB DATE	Apr 71
NOTE	15p.
EDRS PRICE	EDRS Price MF-$0.65 HC-$3.29
DESCRIPTORS	*Computer Programs, Factor Analysis, Models, *Reliability, *Research Methodology, *Scores, *Simulation, Statistical Analysis, Test Reliability

ABSTRACT

A major problem in the research concerning distributional and other properties of reliability coefficients has been the non-existence or inaccessibility of adequate test data for use in empirical verification of hypothetical conclusions. The purpose of this paper is to develop a technique for the simulation of test item scores through the use of computers. The development is relatively straightforward, being based on the principal ideas of factor analysis and on the use of factor loading matrices. Since the researcher determines the factor structure of the tests he simulates, sets the nature of the ability distribution of his simulated subjects, and has direct control over the reliability coefficient for the "population" from which he "draws" his tests, the procedure allows considerable flexibility. The experimenter can easily simulate discrete or dichotomous scores if he desires. Use of the technique does not require advanced programming skill. The method is explicated and the computer program, along with an illustration of its use, is included. (DG)

ED052252

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY

A PROCEDURE FOR THE SIMULATION OF TEST ITEM SCORE DISTRIBUTIONS

Robert Mendro
Department of Educational Psychology
University of Illinois

0. Introduction

A major problem in the research concerning distributional and other properties of reliability coefficients has been the non-existence or inaccessibility of adequate test data for use in empirical verification of hypothetical conclusions. Several recent studies have suffered from such a lack of test results. (Feldt, 1965; Penfield, 1968)

In general, the assumptions underlying the development of various reliability measures do not correspond to the conditions which exist in actual testing situations. Hence, the researcher, in some instances, may be able to obtain data which satisfy some of the assumptions employed in the development of the coefficient which he is investigating, but which may leave other assumptions unsatisfied. Or, all too often, he may find that data which is even partially satisfactory is unavailable or is available only in restricted quantities and is difficult to obtain.

For example, a researcher investigating the properties of coefficient alpha needs data in which both subjects and items have been randomly sampled from infinite sized populations. A person with data available from a national testing bureau or some similar type of organization may be able to simulate random selection of subjects but rarely (if at all) could he realistically simulate random selection of items. A researcher without access to the results of a widely administered test could not

TM 000 658

adequately simulate selection of either items or subjects.

The purpose of this paper is to present a technique for the simulation of test item scores through the use of digital computers. The technique has several practical advantages. First, its development is relatively straightforward. The method is based on elementary use of factor loading matrices, hence, the only specialized background required is a basic understanding of the principle ideas of factor analysis. The procedure allows a considerable amount of flexibility in the conditions underlying the simulated test which it produces. The researcher determines the factorial structure of the tests he simulates and can control the nature of the ability distribution of his simulated subjects. Also, he has direct control over the reliability coefficient for the "population" from which he "draws" his tests. Finally, the implementation of the technique does not require advanced programming skill. The only intermediate level features employed in the sample FORTRAN program presented in the latter part of this paper are subroutine calls and simple matrix operations. (Note that strictly speaking the subroutines are desirable features but not <u>necessary</u> features of those programs. They could have been written into the main program.)

1. Rationale for the Simulation Procedure

The simulation procedure is based on the following rationale. Assume we are given a test score distribution for a test with i items and n subjects in the form of an i x n matrix. This matrix can be broken down into the sum of two i x n matrices, the first being a matrix of "true scores" for the n subjects on the i items and the second being a matrix of error components. The i x n matrix of "true scores" can be further broken down into the product of two matrices. The first is an i x f matrix of factor loadings where f is the number of factors obtained from a given factorization of the test. Each row of this matrix contains f loadings, one for each factor, with one row corresponding to each item. The second matrix is an f x n matrix of factor scores for the n subjects. Each column of this matrix contains f scores with one column for each subject. Hence, we will assume that any test score matrix (matrix T) is composed of the product of an item factor loading matrix (matrix L) times a subject factor score matrix (matrix S) plus an error matrix (matrix E). In symbols this is represented as

$$(1.1)\quad T_{i \times n} = L_{i \times f} \times S_{f \times n} + E_{i \times n}.$$

Now, assuming the model represented in equation 1.1, each of the component matrices (L, S, and E) which make up a test score matrix (T) can be simulated. The basis of the simulation procedure will be to simulate a factor loading matrix, a factor score matrix, and an error matrix for each test we desire to create. Then by post-multiplying the first matrix by the second and adding the third to this product we can produce a simulated test score matrix.

2. Technical Outline of the Simulation Procedure

In an actual simulation a reasonable outline of the steps necessary to implement the procedure would be as follows:

a) The researcher must determine the sample size (n), the number of items for each test (i), the number of factors in the underlying structure (f), the type of sampling procedure to be used for items and for subjects, i.e. whether items or subjects will be fixed or sampled randomly from a population, and the relative distribution of ability in the subjects.

b) Next, if a specific factor loading matrix is not being used, as would be the case if items were randomly selected, the size of the loadings in the i x f factor loading matrix must be determined. Limits must be established for the range of the loadings on each factor. For example, the investigator might want a general factor appearing in every item with a loading ranging between .2 and .4. Or he might desire a specific factor for a quarter of the items which had a loading between .35 and .48, etc. (It should be noted that the sum of the squares of the maximum value of each loading should be less than or equal to 1.) Since actual test data is not always very "neat" factorially, the investigator might find it desirable (or realistic) to add items with relatively low loadings on all factors. It will be shown later that these items can have relatively large error components as a result.

Once he has determined these limits the researcher can choose one of two general approaches to sampling items. He may generate a

new matrix of factor loadings for each simulated test. This would involve sampling a random number for each loading, with the number falling within the specified limits. Or if time was a factor or the study was exceptionally large, he could generate a relatively large pool of sets of item loadings (say 100 to 1000 depending on the study) and randomly sample sets of loadings from this pool. In the example given in section 3 of this paper the latter procedure is used.

c) Assuming subjects are to be randomly sampled, it is necessary to determine the characteristics of the population of subjects. This is done by determining the method of constructing factor scores for the subjects. A direct method of obtaining scores with an approximately normal distribution would be to assign factor scores by random sampling from a uniform distribution. When these scores are multiplied by the factor loadings and added together, the resulting scores will usually tend toward normality. A method which could be employed to obtain skewed distributions of subject ability would be to sample scores for a proportion of subjects from a uniform distribution and the scores for the rest from a truncated asymptotic distribution. For example, a negatively skewed distribution of ability could be obtained by sampling scores for 30% of the subjects from a uniform distribution ranging from .5 to 1.5 and sampling the scores for the remaining 70% of the subjects from a unit normal distribution which has been truncated at .6 or .7 (i.e. factor scores above .6 or .7 are rejected and sampling continues until a score below the limit is obtained.)

d) The researcher must now determine the error scores. After the item loading matrix is sampled from its population, an error loading for each item is computed. This is done by squaring the loadings for an item, summing then, subtracting this value from one, and taking the square root of the result. Repeating this process for each item, an i x 1 vector of error loadings is obtained, one loading for each item.

To obtain the error score matrix, an i x i diagonal matrix is contructed with the vector of error loadings forming the diagonal of this matrix. The diagonal matrix can be then post-multiplied by an i x n matrix of randomly selected components from a uniform distribution to obtain the i x n matrix of error scores. Note that more complex error structures can be devised by distributing the error loading vector for the items into several vectors, each error vector then being multiplied by a specific matrix of random components designed to represent some type of error component in a subjects response to an item.

Since the error loading for an item is determined by the size of the factor loadings, it is obvious that the researcher can directly control the size of the error loading by controlling the limits assigned to each factor loading. Hence, he can directly control the proportion of error variance in his simulated tests as well.

e) The experimenter, if he desires, can simulate discrete or dichotomous scores fairly easily. If he wants to simulate discrete scores all he need do is take the continuous scores produced above,

multiply by an appropriate factor to achieve the proper variance in his scores, and force his data from floating point to fixed point numbers. To obtain dichotomous items all he need do is calculate a cutoff score for each item which he then compares to each subject's score on the item and then assigns a 0 or 1 for the item score depending on whether the cutoff is larger or smaller than the subject's score.

It can be seen from the outline of the procedure above that the researcher has a considerable amount of freedom to determine the conditions under which his simulated tests are produced. As a result, the accuracy of the procedure is limited only by the accuracy of the model which the investigator chooses to employ in building his tests.

3. A Sample Program and Data

A sample program is presented in this section. Running times on the machine used are given and a sample score matrix is shown. This program is a sample program designed to illustrate the implementation of the procedure described in section 2 of this paper. It has been run successfully on the machine mentioned. However, no claim is made that it is necessarily a reasonable adaptation of the procedure for any other problem or that it will run successfully on other machines.

The program was used in a study conducted by the author (Mendro, 1970). The main program used in the study is omitted since most of it is irrelevant to this paper. Instead a small main program written to call the various subroutines involved is included.

The object of the study was to produce stratified-parallel tests to investigate distributional properties of the stratified-alpha generalizability coefficient. The tests produced had three strata, each stratum with an equal number of items. The number of items per test then was three times the number of items per stratum. The factor model underlying the tests was as follows. There was a general factor which loaded between 0.17 and 0.34 on all items. There were two stratum factors for each stratum. The limits for the loadings of these factors were 0.22 to 0.44, 0.19 to 0.38, and 0.16 to 0.32 for their respective strata. When an item was chosen for the particular stratum involved, the loadings on the remaining two strata were allowed to range between 0.0 and 0.1. Hence, typical rows of the factor loading matrix for the three strata looked like those listed in Table 1.

Table 1: Sample Item Factor Loadings

	General Factor	Stratum 1		Stratum 2		Stratum 3	
Stratum 1	.193	.395	.399	.018	.081	.088	.068
Stratum 2	.247	.064	.094	.365	.265	.003	.002
Stratum 3	.218	.028	.067	.063	.031	.300	.235

To save running time a pool of loadings was constructed with 300 rows, 100 for each stratum. When a particular test was being constructed, the rows for that test were randomly sampled from this pool, with replacement.

Factor scores for subjects were generated by randomly sampling from a uniform distribution between the limits 0.0 and 1.0.

The item scores for each subject were dichotomized by comparing them to cutoffs which were associated with each item. The cutoff was calculated by adding the factor loadings and error loading for the item together, dividing by two, and then adding a random number in the range -0.33 to 0.33.

The entire study was run on the CDC 6400 computer of the Graduate School Computer Center at the University of Colorado. In all runs the number of items per stratum was 10 giving a total of 30 items per test. The running times given below include the time needed to calculate the value of stratified alpha for each test as well as the time needed to construct the scores, hence they will be somewhat larger than should normally be expected. Running time is given for the central processing unit in seconds for each set of 1000 tests generated. Table 2 gives these results.

Table 2: Run Times

Number of Tests	Number of Subjects	Running Time
1000	15	159.520
1000	30	297.487
1000	60	574.752
1000	90	850.819

The programs for generating the sample tests are listed in Figures 1 - 3. A few technical details concerning the programs are listed below:

1) Formats have been eliminated to save space.

2) The variable POP holds the pool of item factor loadings.

3) The variable TEMP holds the item error loadings.

4) The variable CUT holds the item cutoff scores.

5) The variable SC holds the constructed score matrix for a given stratum.

6) SELECT is the subroutine which constructs the sample tests.

7) DICOT is the subroutine which dichotomizes the score matrix.

Part of a sample score matrix produced by these programs is given in Figure 4.

As was noted above, the programs are not very difficult to understand and persons with a knowledge of intermediate level FORTRAN should be able to implement the technique in their own investigations.

```
      DIMENSION POP(7,100,3),TEMP(100,3),CUT(100,3),SC(10,90)
      COMMON POP,TEMP,CUT,SC
      READ 2, POP,TEMP,CUT,NP
C NP=NUMBER OF PERSONS
      DO 10 I=1,1000
      DO 20 J=1,3
      CALL SELECT (J)
      CALL COMPUTE (J)
C COMPUTE IS A SUBROUTINE (NOT INCLUDED)
C DESIGNED TO COMPUTE THE MEAN,VARIANCE,
C AND OTHER VALUES FROM THE STRATUM SCORE
C MATRIX PRODUCED BY SELECT.
   20 CONTINUE
      CALL ALPHA
C ALPHA IS A SUBROUTINE (NOT INCLUDED) WHICH
C COMPUTES THE VALUE OF STRATIFIED ALPHA
C AND PRINTS IT OUT.
   10 CONTINUE
      END
```

Figure 1: Main Program

```
      SUBROUTINE SELECT(J)
      DIMENSION POP(7,100,3),TEMP(100,3),CUT(100,3),SC(10,90),
     A  FITEMS(10,7),QZ(10),OFF(10),RANMAT(7)
      COMMON POP,TEMP,CUT,SC
      DO 20 I=1,10
C I IS THE INDEX FOR ITEMS.
      NR=RANF(0)*100
C RANF IS A UNIFORM RANDOM NUMBER GENERATOR
C (ON LINE) IN THE RANGE 0.0 to 1.0
      NR=NR+1
      DO 20 K=1,7
      FITEMS(I,K)=POP(K,NR,J)
C FITEMS HOLDS THE SELECTED LOADINGS
      QZ(I)=TEMP(NR,J)
C QZ HOLDS THE CORRESPONDING ERROR LOADINGS
      OFF(I)=CUT(NR,J)
C OFF HOLDS THE CORRESPONDING CUTOFFS
   20 CONTINUE
      DO 40 L=1,90
      DO 35 MD=1,7
      RANMAT(MD)=RANF(0)
   35 CONTINUE
      DO 30 I=1,10
      SC(I,L)=0.
      DO 10 K=1,7
      SC(I,L)=SC(I,L)+FITEMS(I,K)*RANMAT(K)
   10 CONTINUE
      SC(I,L)=SC(IL)+QZ(I)*RANF(0)
   30 CONTINUE
   40 CONTINUE
      CALL DICOT(SC(1,1),OFF(1),10,90)
      RETURN
      END
```

Figure 2: Subroutine SELECT

```
      SUBROUTINE DICOT(SC,OFF,NITEM,NPERS)
      DIMENSION SC(30,90),OFF(30)
      DO 10 I=1,NITEM
      DO 15 J=1,NP
      IF(SC(I,J).GT.OFF(I))GO TO 20
      SC(I,J)=0.
      GO TO 15
   20 SC(I,J)=1.
   15 CONTINUE
   10 CONTINUE
      RETURN
      END
```

Figure 3: Subroutine DICOT

	PERSON							
ITEM	1	2	3	4	5	6	7	8 . . .
1	0	1	1	0	1	1	1	0 . . .
2	1	1	1	0	1	1	0	1
3	0	1	1	1	1	1	0	1
4	1	1	1	1	1	0	1	1
5	0	0	0	0	0	0	0	0
6	0	0	1	0	0	1	0	0
7	0	1	1	0	0	0	0	1
8	1	1	1	0	1	1	0	0
9	0	1	1	1	1	1	1	0
10	0	1	1	0	0	1	0	1 . . .

Figure 4: Part of a Sample Score Matrix

4. Summary

The procedure as outlined above presents a practical method of simulating test score matrices. It can be used in an elementary fashion or it can be adapted to very complex test models. It serves a useful purpose for those researchers who cannot have access to large amounts of real test data.

Naturally, real test data is preferable to simulated data, no matter how good or accurate the simulation. However, in practical situations the amounts and types of data that can be obtained are usually restricted. Hence, simulated data can be exceptionally useful and an acceptable substitute for the "real thing".

Bibliography

Feldt, Leonard S. "The Approximate Sampling Distribution of Kuder-Richardson Reliability Coefficient Twenty," Psychometrika, 30: 357-370; September, 1965.

Mendro, Robert L. "An Approximation to the Sampling Distribution of the Generalizability Coefficient for Stratified-Parallel Tests." Unpublished Master's thesis, University of Colorado, 1970.

Penfield, Douglas Alan. "An Empirical Investigation of the Approximate Sampling Distribution of Kuder-Richardson Twenty." Berkeley: University of California, 1968. (Multilith.)